IN-61
198086
33 P

NASA Contractor Report 177627

Superpatch

Jin J. Chou and Francis Enomoto

(NASA-CR-177627) SUPERPATCH (Computer Sciences Corp.) 33 p

N94-21489

Unclas

G3/61 0198086

CONTRACT NAS2-12961
October 1993

National Aeronautics and
Space Administration

NASA Contractor Report 177627

Superpatch

Jin J. Chou

Computer Sciences Corporation
P.O. Box 390757
Mountain View, CA 94039

Francis Enomoto

Ames Research Center
Moffett Field, CA 94035-1000

Prepared for
Ames Research Center
CONTRACT NAS2-12961
October 1993

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

PAGE ii INTENTIONALLY BLANK

Contents

SUPERPATCH*

Jin J. Chou
Computer Sciences Corporation
P.O. Box 390757
Mountain View, CA 94039

Francis Enomoto
Aerodynamics Division
NASA Ames Research Center
M/S 227-2
Moffett Field, CA 94035

October 20, 1993

Abstract

This document describes a design model geometry file format with topology information used for model geometry exchange in Computational Fluid Dynamics (CFD) grid generation. The first part of this document describes the reason why this file format is necessary and what we want to achieve through the file format. The second part provides the information stored in the file format. The purpose of the publication of this document is for broad dissemination of this file format and for the wide participation of testing.

*Christened by Dr. Michael George.

PAGE 2 INTENTIONALLY BLANK

1 Introduction

This document describes a design model geometry file format with topology information used for model geometry exchange in Computational Fluid Dynamics (CFD) grid generation. The document is organized as follows:

Section 2 describes the reason why this file format is necessary and what we want to achieve through the file format.

Section 3 provides the information stored in the file format.

People who are interested in only the file format may want to skip Section 2.

At the time of writing, the file format is a proposed format. Testing of this file format both in data exchange and in integration with grid generators is underway. The file format will remain as a proposed format until the tests are done and their results are fully evaluated.

2 Objective and Domain Selection

2.1 Introduction

A particular model representation used in a Computer-Aided Design (CAD) system may not be the desired one for Computational Fluid Dynamics (CFD) grid generation. Most of the time, CAD surfaces are not aligned with the desired computational domains, i.e., the computational domains may have different boundaries and directions from those of the CAD surfaces. Also, with different design approaches, the same CAD model may be represented by radically different types of surface arrangement. Therefore, it is desirable for the grid generation process to grid on a CAD-independent representation, in particular, to grid across multiple CAD surfaces, to grid on trimmed surfaces, and to grid on a part of a surface.

In the Rockwell Automated Grid Generation System (RAGGS) [1], arbitrarily connected Bezier surfaces can be sewed together to form "quilts." Grid generation is performed on these quilts. The sewing is done by keeping the topological information among the Bezier surfaces. Since topological information is not kept between quilts, no gridding across quilts is possible. By utilizing quilts, RAGGS has been able to perform gridding partially independent of the arrangement of the CAD surfaces. In general, by keeping the topological information on the whole model, the grid generation process will be able to provide better gridding capability.

IGES [3] is the most widely used product data exchange standard. Manifold Boundary Representation (B-Rep) is included in the Grey Pages of IGES Version 5.1. B-Rep contains the full topological information for a model. The representation is under testing currently and will move into the main body of the IGES document in the next release (Version 6.0). However, B-Rep models and surface models belong to two different classes. It is expected that geometry exchange without topology will be used for years to come.

The data in the current NASA IGES (NIGES) files [2] provides surface geometry only and is called *surface models*. Such data does not contain the explicit topological information among the surfaces. In the NIGES document, to include topological information is listed as the next-level extension of the data exchange method. Several comments from the industrial review of the NIGES document also suggested the need for including topology in data exchange.

2.2 Requirement

The goal of the Superpatch definition is to define a CAD-independent representation, called "Superpatch," to enable the user to perform the so called "CAD-independent gridding," specifically,

- to grid across multiple CAD surfaces,
- to grid on trimmed surfaces,
- to grid on a part of a surface, and
- to perform gridding independent of CAD surface arrangement and surface parametrization.

To reach this goal, two things have to happen. First, a model representation containing the necessary information needs to be defined. Second, grid generation software has to utilize this information and provides options for the users to perform CAD-independent gridding. It is clear that full topological information is the key to the CAD-independent representation and gridding. Hence, the Superpatch definition calls for a representation with full topological information.

In addition, the representation will serve better if the following requirements are met.

1. To be able to be converted to and from a NIGES surface model.

   The NIGES surface model does not include explicit topology information. We need to derive the topology information from the surface model by examining the surface geometry. The representation must allow its topology to be derived from the surface geometry.

   It is also desirable to convert the representation to a NIGES surface model for data exchange. Hence, the representation must be able to be converted to a surface model.

2. To be able to be converted to and from the IGES format and still keep all the topological information.

   Since IGES is the most popular data exchange format for geometry design and all CAD systems provide IGES capability, it is desirable that the representation can be converted to and from the IGES B-Rep format.

2.3 Task Objective

The objective of the task is to define

- a file data exchange format
- containing only CAD surface definition information (with explicit topology), and is used for
- exchange design models among CAD systems and grid generators.

Information regarding surface domain partition is excluded because this information does not belong to model geometry and should be recorded and exchanged along with grid information, i.e., along with a full analysis model, see discussion in Section 2.5.

2.4 Terminology

This section describes several terms used in the rest of this document. If you are familiar with the terms, you can skip this section.

manifold A model is a manifold (or two-manifold) if any point p on the surface of the model and the points immediately surrounding p is homeomorphic to a two-dimensional disk. Otherwise, the model is called non-manifold. In a manifold model, only two surfaces can meet along a common boundary. A manifold model is shown in Figure 1. Two non-manifold models are shown in Figure 2. The top model has four surfaces incident along a line which makes the model non-manifold. The bottom model contains a dangling surface meeting with two other surfaces.

parametric surface In this document, a surface (parametric) is a two-dimensional set of connected points in three-space, which is also called the *model space* (see Figure 3). A surface has a domain space which is a rectangle in R^2. The surface is defined by a map from its domain space to the model space. The map is called the *surface map*, $S(u, v)$.

Each curve in the domain space, called a *parameter space curve*, corresponds to a curve on the surface in the model space, called a *model space curve*. The model space curve is obtained by mapping the corresponding parameter space curve through the surface map.

sufficient A representation is sufficient if and only if all the adjacency information can be derived without performing geometric comparisons. That is, all the adjacency information is retained by the pointers in the representation.

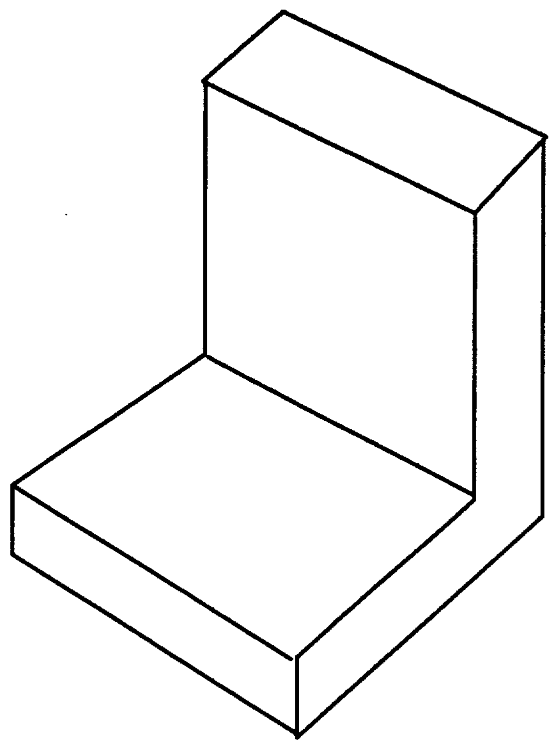

Figure 1: A simple manifold model

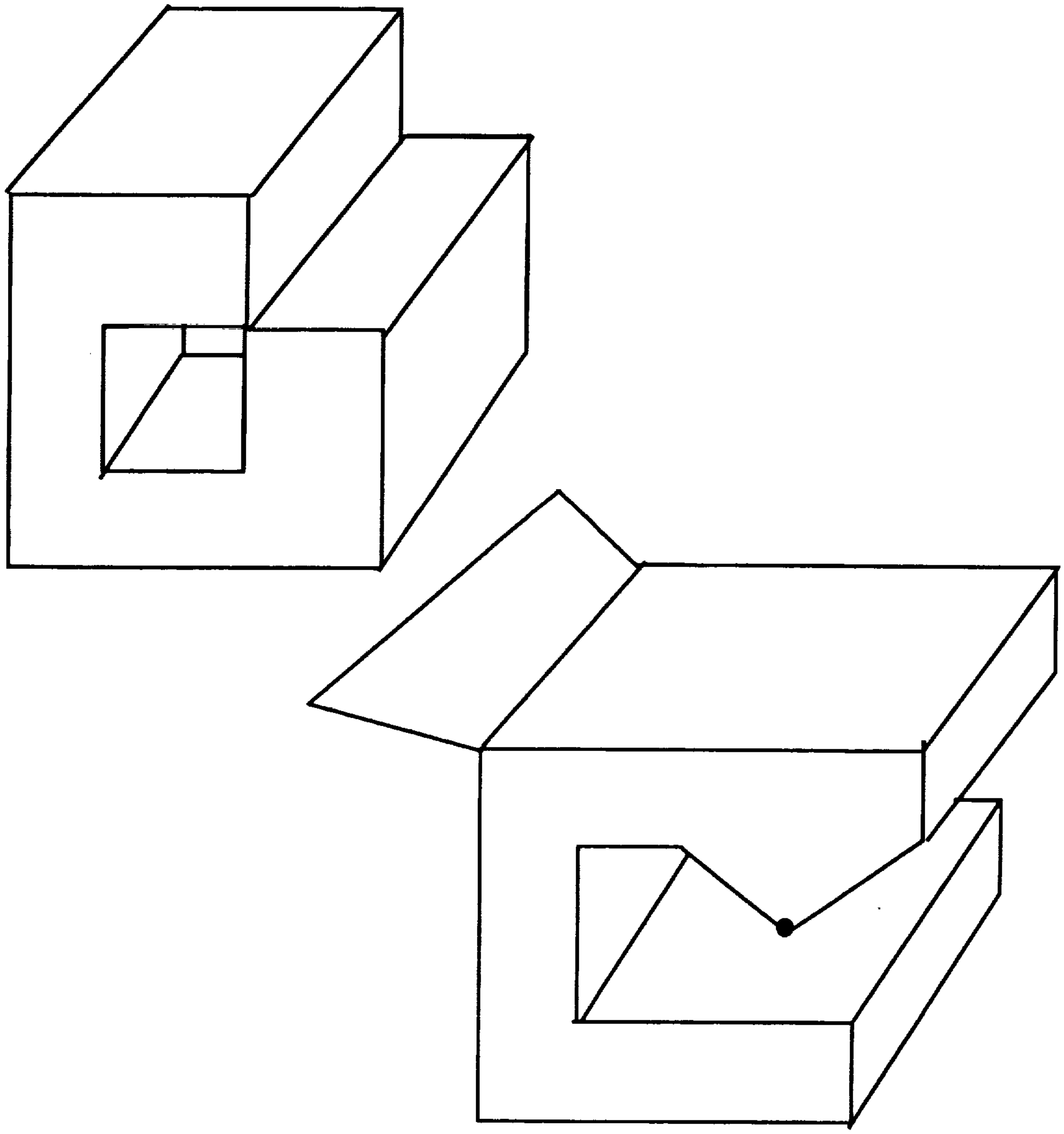

Figure 2: Non-manifold Models

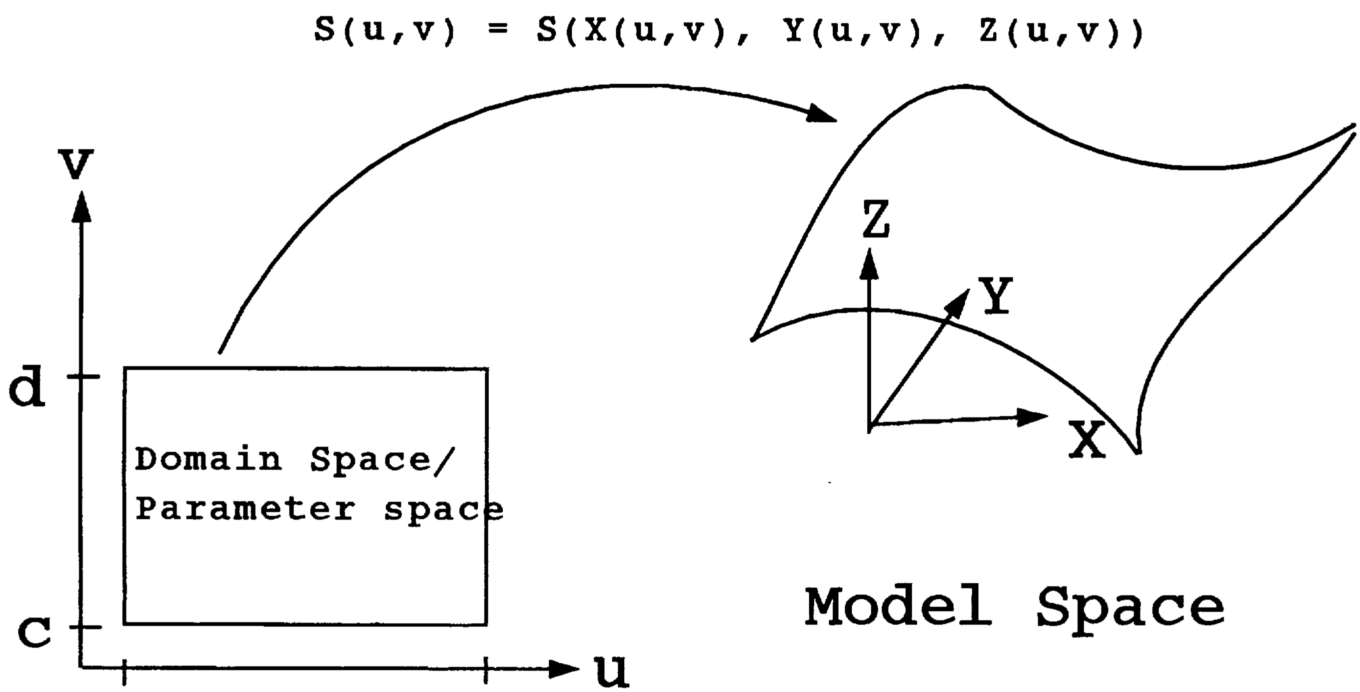

Figure 3: A parametric surface

2.5 Design Model/Analysis Model

During a design process, the designer needs to consider very wide aspects of a product. In the case of a vehicle design, aerodynamic responses, structural strength, functional requirements, and maybe even aesthetic views are due consideration. During the design process, geometry is created to fit all these needs. However, a piece of geometry which is crucial in the structure area may be meaningless in a CFD analysis. Therefore, additional simplifications of a design model are usually carried out to weed out the unnecessary complexity in the design model before the grid generation process. During this simplification process, the design model is converted to an early stage of an analysis model. An example of an early-stage analysis model is a mean surface model. Even within CFD, different analysis methods and different scientists may arrive at different analysis models. Note that during a conceptual design stage, an early-stage analysis model may also be constructed directly without a design model. Commonly the development of the early-stage analysis model is done either in a CAD system or in grid generation software.

After the early-stage analysis model is obtained, further development of the analysis model is done in grid generation software by discretizing the analysis domain. A full analysis model may contain blocking structures, grid distribution, and boundary conditions. Almost all the design models are manifold models. Definitely, all the full analysis models are non-manifold models.

The early-stage analysis models may be manifold or non-manifold. In some cases, the simplifications that derive the early-stage analysis model render the model non-manifold. When an early-stage analysis model is non-manifold, usually, only few locations on the model are non-manifold. The models are in general manifold except at these few locations.

2.6 Possible Domain

Two classes of topological information are commonly used for model representation: Manifold B-Rep (MBRep) and Non-Manifold B-Rep (NMBRep). MBRep is capable of representing manifold models. NMBRep is capable of representing manifold geometry, non-manifold geometry, and wire-frame geometry in one model. Hence, MBRep is a subset of NMBRep. MBRep is adequate in representing the design models that occur in common aerospace designs. In addition to design models, NMBRep is capable of representing analysis models, for both early-stage analysis models and full analysis models.

As stated earlier, MBRep is in IGES. On the other hand, it is unlikely that NMBRep will become part of IGES. Whether NMBRep will be part of STEP (the successor to IGES) remains unknown. Even if it will, it is unlikely for that to happen within the next three years.

Since NMBRep can represent both design models and analysis models, it is best to be used as the basic representation scheme for data exchange of full analysis models or as the internal data representation for grid generation software. By using this representation, grid generation software can avoid representation conversion and information duplication during the various stages of grid generation.

The above discussions and some other properties of the MBRep and NMBRep are listed in Table 1.

	MBRep	NMBRep
Complexity	Less Complex	Much More Complex
Design Model Rep.	Adequate	Adequate
Early-Stage Analysis Model Rep.	Adequate Mostly	Adequate
Full Analysis Model Rep.	Inadequate	Adequate
IGES Data Exchange	Supported	No Support
Support Grid Gen. Software Dev.	No	Yes

Table 1: Comparison of MBRep and NMBRep.

2.7 Discussion on Domain Selection

The choice of the representation for Superpatch depends on its intended use. From Table 1 we found that

1. if the Superpatch definition is used only for data exchange of design models, MBRep is the most suitable representation;
2. if the Superpatch definition is used for data exchange of the early-stage analysis model, MBRep is also the most suitable representation in most cases.

   In rare cases when the early-stage analysis model is non-manifold, the explicit connectivity information on the non-manifold edges will be lost during the data exchange. However, since most of a non-manifold early-stage analysis model is manifold, most topological information about the model can be exchanged with MBRep. The grid generator that receives such data needs to re-derive the non-manifold connectivity information only if the non-manifold connectivity information is used by the grid generator. To re-derive the connectivity information, the grid generation can either derive the information from the geometric data or ask the user to provide the information.

For both design models and manifold early-stage analysis models, the grid generation software will receive the model geometry in MBRep and perform grid generation on MBRep directly or on some internal representation converted from MBRep. The results from the grid generation have to be stored in some other internal representation (essentially a non-manifold representation).

NMBRep would be the most suitable representation, if Superpatch definition were used as the internal representation for grid generation software or as the data exchange mechanism for full analysis models.

2.8 Conclusion

From the above discussion and the objective of Superpatch, we conclude that MBRep is the best representation for Superpatch.

In order to make the file format useful, CAD systems have to produce files in this format. Currently, IGES is the most popular file format supported by nearly all CAD systems. We conclude that the goal of Superpatch is best achieved by insuring the file format is closely aligned with the IGES B-Rep file format.

3 File Information Description

3.1 Introduction

This section describes all the entities in the Superpatch file format, which is based on IGES B-Rep, including Manifold Solid B-Rep Object in IGES 5.1 and the Open Shell Entity, which is still a Request For Change ("To be added into IGES V6.0").

The primary goal of this document is to allow people who are not familiar with IGES to understand what is in IGES B-Rep. However, this is not a complete description of the IGES *file format.* People who engage in writing an IGES pre- or postprocessor should read the IGES document and fully understand the intricacy of the file format. This document should contain enough information for those whose only purpose is to *utilize* the data from an IGES file.

Almost all of the descriptions below are direct quotes from IGES V5.1. Note that in Superpatch format, limited geometry is allowed (B-splines only). In addition, the parametric space curves in edge-uses are mandatory instead of being optional as in IGES B-Rep.

3.2 Topological Entities

The following topological entities are in IGES B-Rep and Superpatch:

```
IGES Entity Type Number        Entity Name
-----------------------        ------------
186                            Manifold Solid B-Rep Object
```

514,Form 1	Shell (Open Shell)
,Form 2	Shell (Closed Shell)
510	Face
508	Loop
504	Edge List
502	Vertex List

3.3 Geometry Entities

IGES B-Rep includes 21 geometry entities. However only B-splines are included in Superpatch:

IGES Entity Type Number	Entity Name
128	Rational B-Spline Surface
126	Rational B-Spline Curve

Description of the B-spline entities can be found in [2].

3.4 Run Down of the Topological Entities

Several topological entities use an Orientation Flag (OF) to indicate whether the direction of a referenced entity agrees with or is opposed to the direction of the referencing entity. If the OF is TRUE then the direction of the referenced entity is correct, but if the OF is FALSE then the direction of the referenced entity should be reversed. It can happen that there are several OF's in the chain of entities from the high level referencing entity to the low level referenced entity.

3.4.1 Entity 186: Manifold Solid B-Rep Object

A manifold solid object is a bounded, closed, and finite volume V in three dimensional Euclidean space, R3. V is restricted to be the closure of the interior of V which must be arcwise connected. V can have an arbitrary number of voids (no restriction on the genus of the boundary surfaces).

The Manifold Solid B-Rep Object (MSBO) defines a manifold solid by enumerating its boundary. This boundary may be decomposed into its maximal connected components called closed shells. Each shell is composed of faces which have underlying surface geometry. The faces are bounded by loops of edges having underlying curve geometry. The edges are bounded by vertices whose underlying geometry is the point. Implicit in the representation is a concept of oriented uses of topological entities by containing entities. This allows the referencing entity to reverse the natural orientation of the referenced entity. The natural orientation is derived from the underlying geometry.

Figures 4 and 5 show the hierarchical nature of the Manifold Solid B-Rep Object and the construction of it. The construction of two simple Manifold Solid B-Rep objects can be found in Appendix A.

The Manifold Solid B-Rep Object describes the boundaries of the solid via oriented uses of shells (shell-use). It is the orientation of the use of the shells that defines the volume of R3 which the MSBO is describing.The orientation of the shell-use is determined by the shell-use normal which is either in the same or opposite direction as the shell normal. By convention the direction of the shell-use normal points away from the part of R3 being described (away from the solid). One shell, the outer, must completely enclose all the other shells and only the outer shell may enclose a shell.

The Manifold Solid B-Rep Object points to one or more closed shells.

```
File Data
---------
- Pointer to the outer shell
- OF of the shell w.r.t. its underlying faces
- Number of void shells, or zero
- Pointer to the first void shell
- OF of the first void shell
     .
     .
     .
```

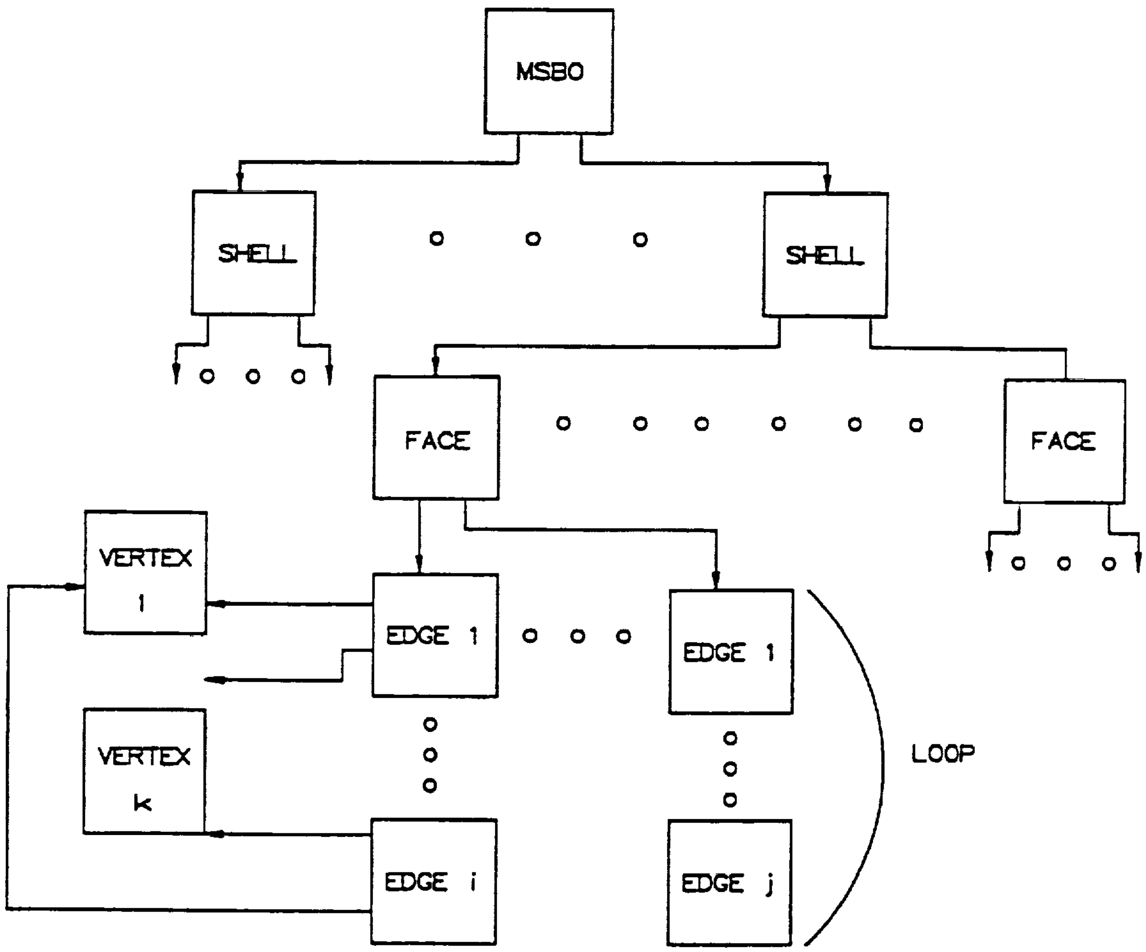

Figure 4: Hierachical nature of the MSBO

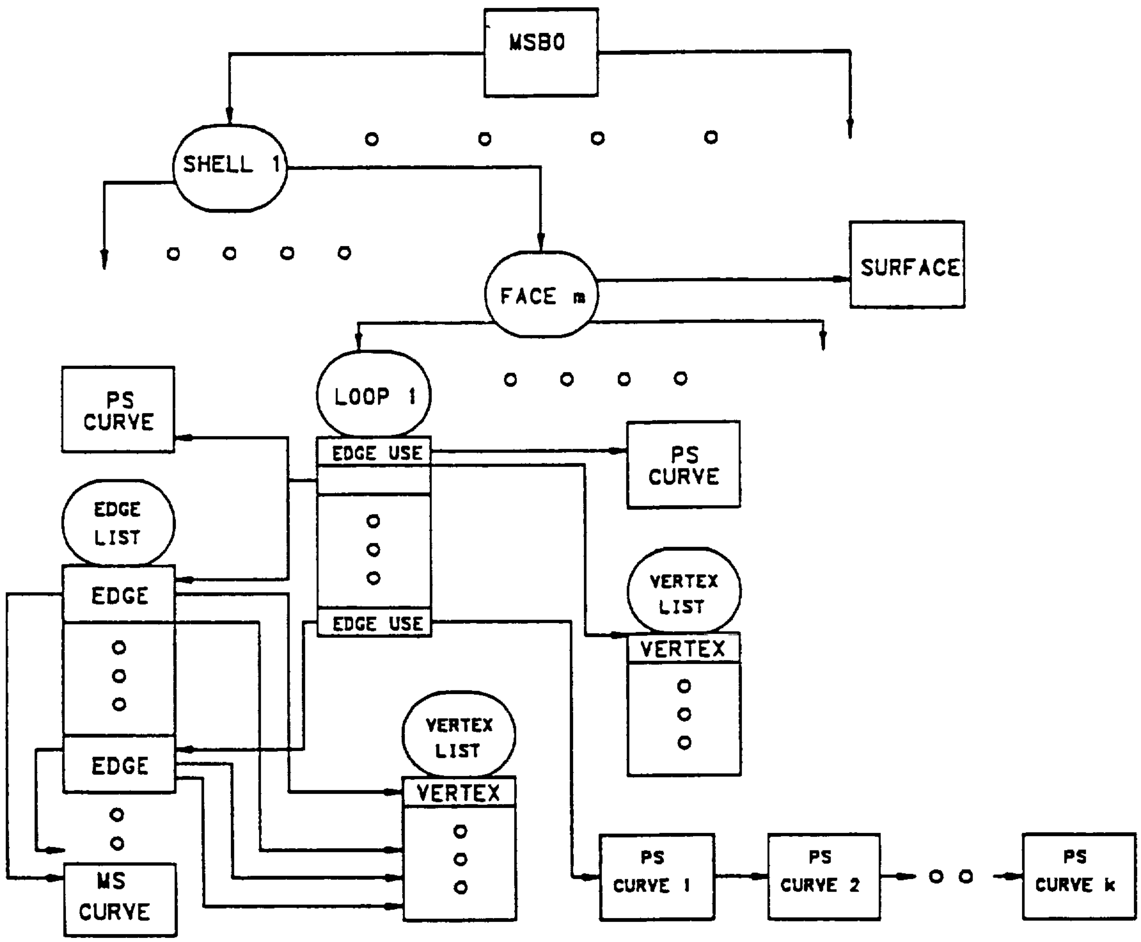

Figure 5: Construction of the MSBO

```
- Pointer to the last void shell
- OF of the last void shell
```

Note: the OF in the file data is a direct quote from IGES V5.1. However, it is very confusing as it is written. It is better written as "- OF of the shell-use in the manifold solid object w.r.t. the shell entity pointed by the pointer to the shell."

Figure 6 shows an MSBO with three shells. The OF of the two inner shells shall be false.

3.4.2 Entity 514: Shell

There are two forms of this entity. Form 1 is Closed Shell. Form 2 is Open Shell.

The shell is represented as a set of edge-connected oriented uses of faces (face-use). The normal of the shell is in the same direction as the normal of its face-uses. The normal of the face-use is assumed to be in the direction of the normal of the underlying surface of the face unless the face-use OF indicates it needs to be reversed. The faces used by the shell are connected to each other only via edges.

Each edge shall be referenced at least once but not more than twice by the loops of the faces of an Open Shell. Each edge shall be referenced exactly twice by the loops of the faces of a Closed Shell.

The topological normal of the shell is the same direction as the normal of any of the surfaces underlying the faces which compose the shell unless the OF associated with the face-use by the shell is false. In this case the direction of the topological normal can be determined by reversing the direction of the surface normal.

- The shell shall be an orientable surface with the same orientation maintained, that is with consistent topological normal.
- The shell must contain at lease one use of a face.

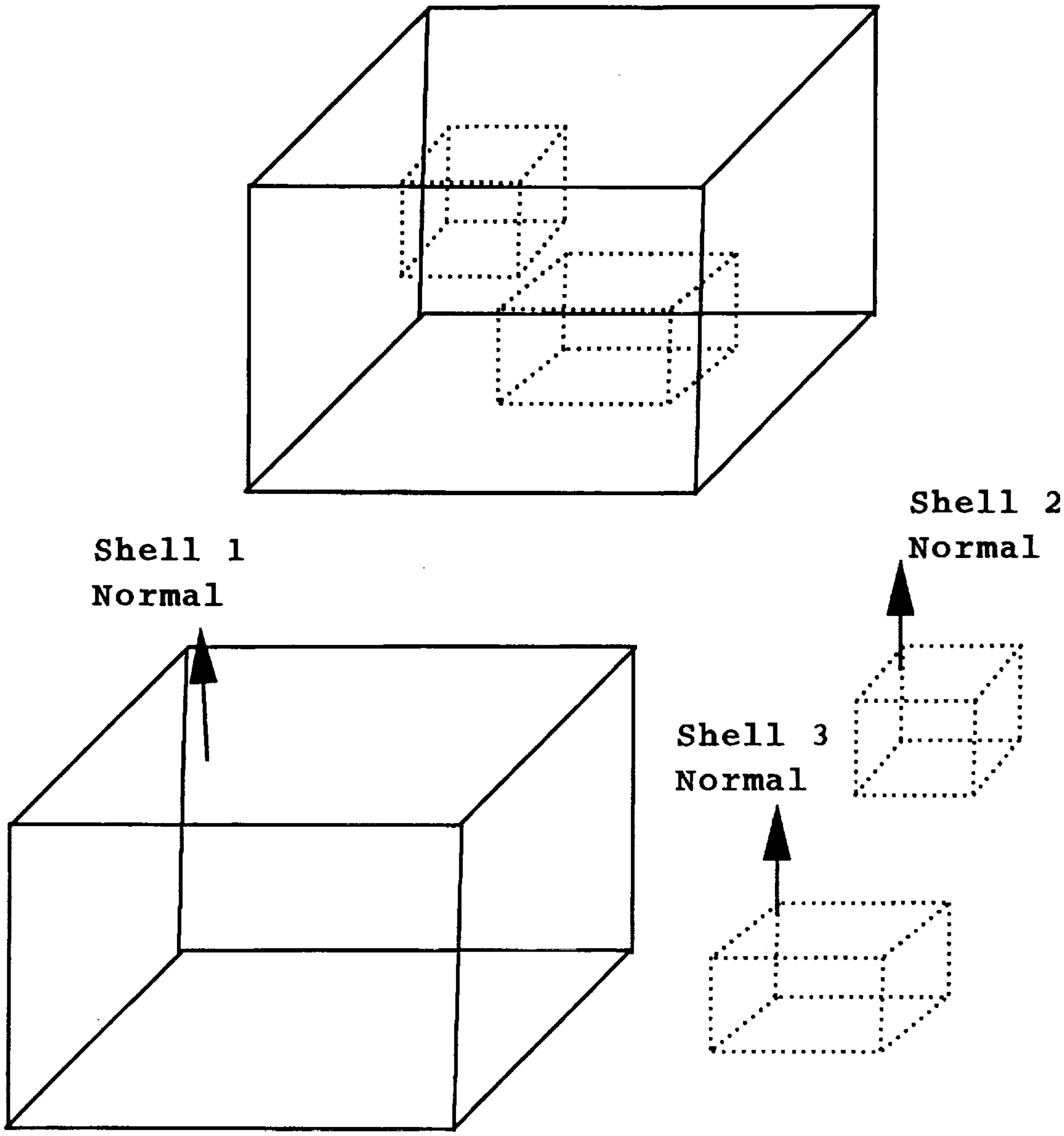

Figure 6: An MSBO with one outer shell and two inner shells.

- Faces in a shell may not intersect themselves or each other except at their edges.

A closed shell is a connected entity of dimensionality 2 which divides R3 into two arcwise-connected open subsets (parts), one of which is finite. The inside of the shell is defined to be the finite region.

The open shell can be thought of as a closed shell with one or more holes punched in it. An example of an open shell can be found in Appendix A.

```
File Data
---------
- Number of faces ( > 0 )
- Pointer to the first face
- OF of the first face w.r.t. the direction of the underlying
  surface
    .
    .
    .
- Pointer to the last face
- OF of the last face
```

3.4.3 Entity 510: Face

Face Entity is a bound (partial) of an arcwise connected open subset of R3 which has finite area. A face, not including its boundary, does not self-intersect and has no handles. The face, F, has an underlying surface, S, and is bounded by one or more loops. If more than one loop bounds a surface, then the loops must be disjoint. The material of the face lies on the left of all the loops bounding the face. See Loop Entity for a definition of left. The normal of a surface, S(u,v), is given by the cross product of the partial derivatives (in the order stated) with respect to u and v. The loops of a face are disjoint.

```
File Data
---------
```

```
- Pointer to the underlying surface
- Number of loops ( > 0 )
- Outer loop flag (True implies that the first loop is an outer loop.
    False implies tht no outer loop. )
- Pointer to the first loop of the face
    .
    .
    .
- Pointer to the last loop of the face
```

3.4.4 Entity 508: Loop

Loop Entity specifies a bound of a face. Typically, a loop represents a connected collection of face boundaries, seams, and poles of a single face (see Appendix A). Its underlying geometry is a connected curve or a single point in R3.

Loop Entity consists of an ordered, repeating construct, the edge-use. This construct consists of either an edge, an orientation, and parameter space curves, or (in the case of a pole) a vertex and a *mandatory* parameter space curve. If the edge-use references an edge, the orientation describes whether the direction of this use of the edge is an agreement with the natural orientation of the edge. An edge-use is only used once in the shell[1]. The loop is represented as an ordered list of edge-uses (EUi, i = 1,n), which has the following properties:

- The terminal vertex of EUi is the initial vertex of EUi+1, i = 1, n-1.
- The loop is closed. This implies that the terminal vertex of EUn is the same as the initial vertex of EU1.
- The orientation of the loop is defined to be the same as its constituent edge-uses.

Let P be a point on the arc of the R3 curve, C, underlying an edge, E. Both P and C lie on the surface S. Let N be the normal vector of S at P. T is a

[1]However, IGES does not provide a mechanism to ensure that an edge-use only appears once in a file.

vector at P whose direction is that of C at P. RT is the vector derived by reversing the direction of T. If the edge orientation flag (OF) is true, then the cross product N X T points to the left of E. If the OF is false, then the cross product N X RT points to the left of the edge.

By convention loops are oriented so that the material of the face they bound lies on the left.

The parameter space curves, Pi, i = 1, n, referenced by an edge-use are in the parameter space defined by the surface, S, underlying the face bounded by the loop containing the referencing edge-use. These curves are assumed to be oriented in the list and oriented such that as the parameter goes from its inital to its final value for each parameter space curve the composition (S o Pi, i = 1, n) produces a composite curve, Ci, i = 1,n, which is coincident with the curve underlying the edge. The orientation of Ci, i = 1, n is in agreement with the orientation of the edge-use.

```
File Data
---------
- Number of edge tuples
- Type of the first edge: 0 = edge; 1 = vertex
- Pointer to the first Vertex List or Edge List Entity
- List Index into the Vertex List or Edge List Entity
- OF of the first edge w.r.t. the direction of the model space curve(s).
- Number of underlying parameter space curves
- Isoparametric flag of the first parameter space curve (True = curve
  is isoparametric on the surface underlying the face which this loop
  bounds)
- Pointer to the first parameter space curve in first edge
     .
     .
     .
- Isoparametric flag of the last parameter space curve
- Pointer to the last parameter space curve in the first edge.
     .
     .
     .
- Type of the last edge: 0 = edge; 1 = vertex
- Pointer to the last Vertex List or Edge List Entity
```

```
- List Index into the Vertex List or Edge List Entity
- OF of the last edge w.r.t. the direction of the model space curve(s).
- Number of underlying parameter space curves
- Isoparametric flag of the first parameter space curve (True = curve
  is isoparametric on the surface underlying the face which this loop
  bounds)
- Pointer to the first parameter space curve in last edge
      .
      .
      .
- Isoparametric flag of the last parameter space curve
- Pointer to the last parameter space curve in the first edge.
```

3.4.5 Entity 504: Edge List

Edge List is a list of edges. An edge represents the topological construct corresponding to a line segment between two vertices. The edge is not closed since it does not contain the vertices which bound it. The start and terminate vertices do not have to be distinct. Edges do not intersect except at their boundaries (i.e., vertices).

Underlying curve geometry in R3 is required. These curves must be continuous and non-self-intersecting in the arc of the curve underlying the edge.

The natural orientation of the edge is in the same direction as its underling curve in R3. The edge is traced from start vertex to terminate vertex as the underlying curve is traced in the direction of increasing parameter value. Each edge should be referenced exactly twice in an MSBO.

The order of the edges in the this edge list is not significant.

```
File Data
---------
- Number of edge tuples in list ( > 0 )
- Pointer to the first model space curve
- Pointer to the Vertex List Entity for the first start vertex
- List Index of the first start vertex in the Vertex List Entity
- Pointer to the Vertex List Entity for the first terminate vertex
```

- List Index of the first terminate vertex in the Vertex List Entity

.

.

.

- Pointer to the last model space curve
- Pointer to the Vertex List Entity for the last start vertex
- List Index of the last start vertex in the Vertex List Entity
- Pointer to the Vertex List Entity for the last terminate vertex
- List Index of the last terminate vertex in the Vertex List Entity

3.4.6 Entity 502:: Vertex List

The geometry underlying a vertex is a point in R3. A vertex is the bound of an edge and can participate in the bounds of a face. There are no default values for the vertex.

The Vertex List Entity contains one or more vertices. Vertex List entity is forbidden to have an associated Transformation Matrix Entity.

The order of vertices in this list is not significant.

```
File Data
---------

- Number of vertex tuples in list ( > 0 )
- X coordinate of the first vertex
- Y coordinate of the first vertex
- Z coordinate of the first vertex
    .
    .
    .
- X coordinate of the last vertex
- Y coordinate of the last vertex
- Z coordinate of the last vertex
```

3.5 Remarks

A couple of remarks follow:

- IGES B-Rep is a *sufficient* description of manifold topology in curve and surface domain. However, for the purpose of saving in file size, IGES B-Rep does not store backpointers from lower level topological elements to higher level topological elements.

  With the simple topological information, some queries to the topology may need to be answered by performing an extensive search through the database with pointer comparisons. For example, at times it is necessary to know the two faces which share a given common edge. To achieve this with the topological information in IGES B-Rep, it is necessary to search through all the faces, with recursion down into all the loops and comparing all the edges, to see if the edge is the same as the given edge. To avoid such extensive search, it may be wise to store a backpointer from an edge to the faces.

- IGES B-Rep does not have explicit *use* entities, for example, no face-use entities or edge-use entities. All the use are stored under its parent entities. For example, the edge-uses are part of a loop entity; the face-uses are part of a shell.

Appendix A: Examples of Manifold Solid B-Rep Objects and Open Shell

Example of an aircraft configuration modeled as an open shell.

In Figure 7 the open shell model is made of two surfaces, one for the wing, and the other for the fuselage. The surfaces are formed by folding their parameter space in the direction of the heavy broken arrows. The model is an open shell since no capping is on either the right end of the fuselage or the tip of the wing. The dashed lines on the leading edge of the wing and the bottom of the fuselage are the silhouette edges of the surfaces.

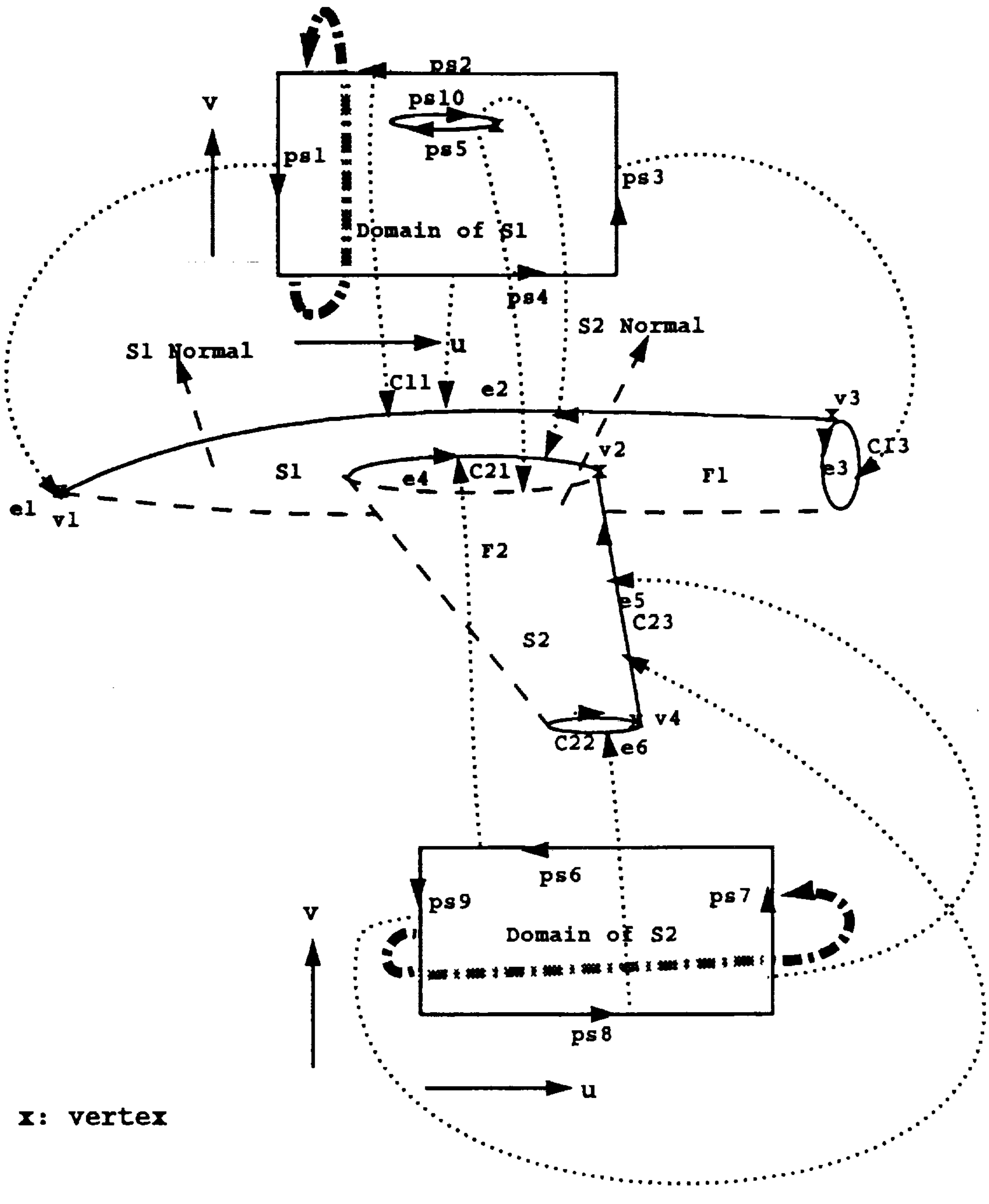

Figure 7: An open shell and the domain spaces of the surfaces.

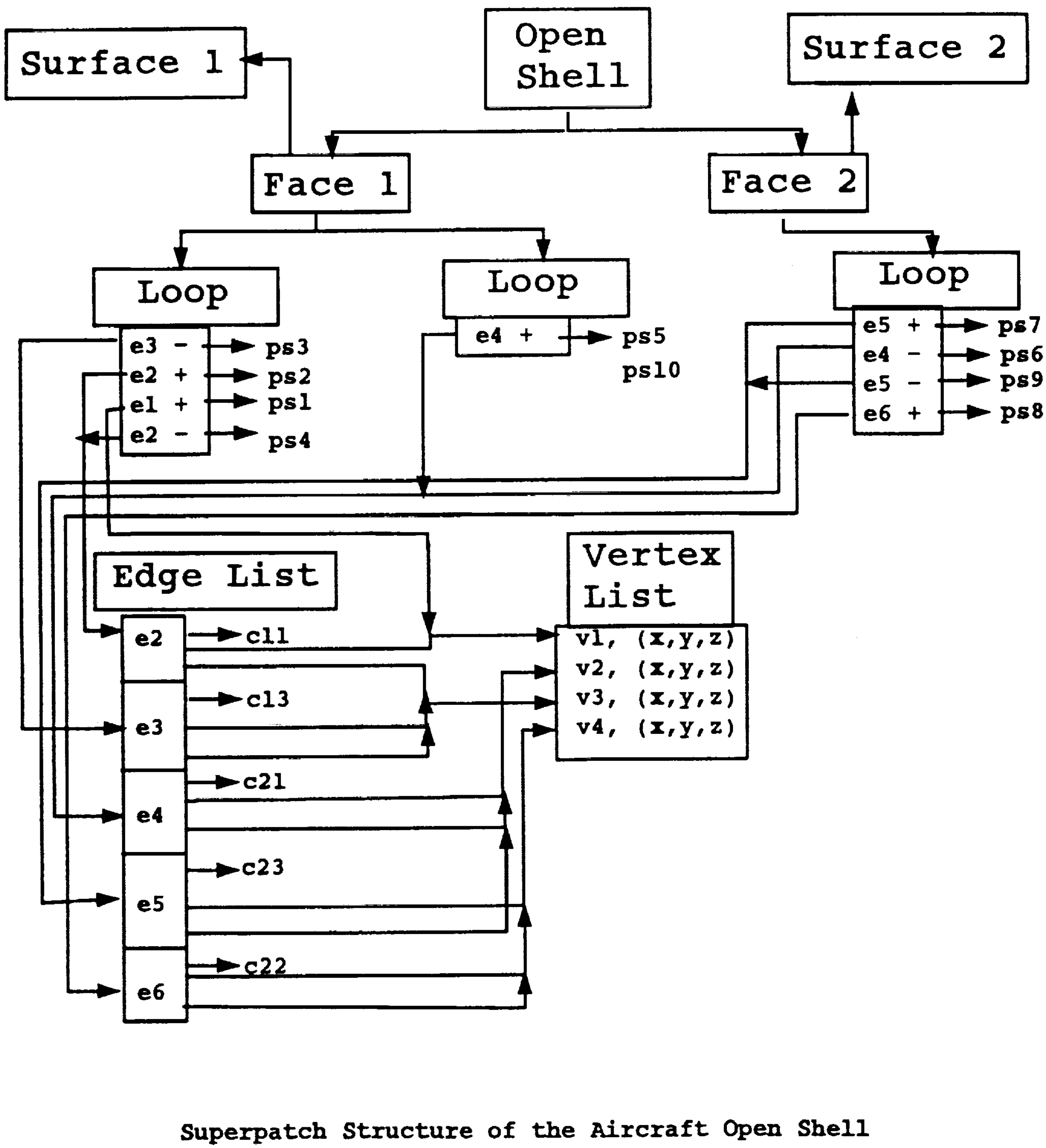

Figure 8: Superpatch structure of the aircraft open shell.

Examples of MBSO:

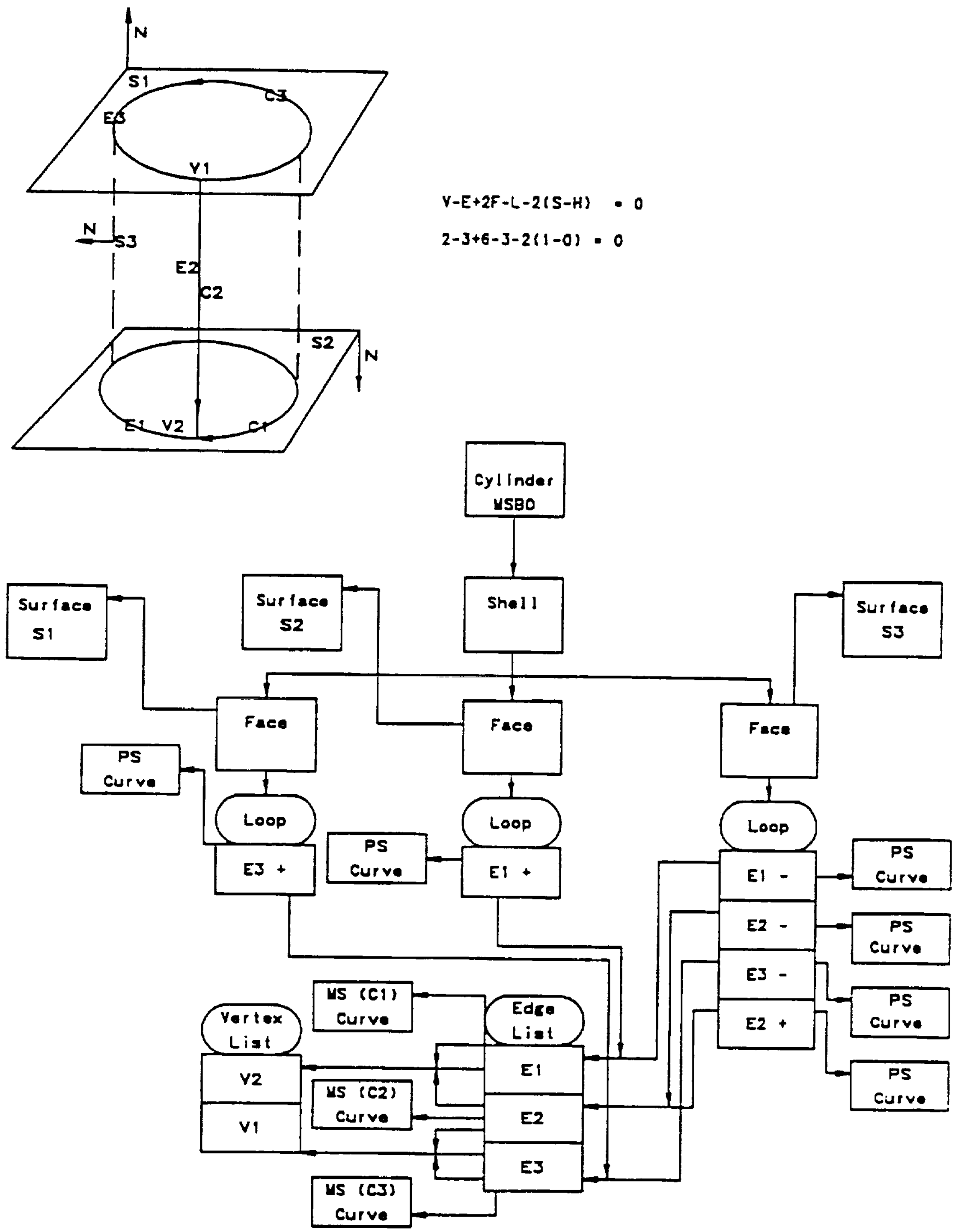

Figure 9: One of the possible MSBO representations of a cylinder with capping planar surfaces.

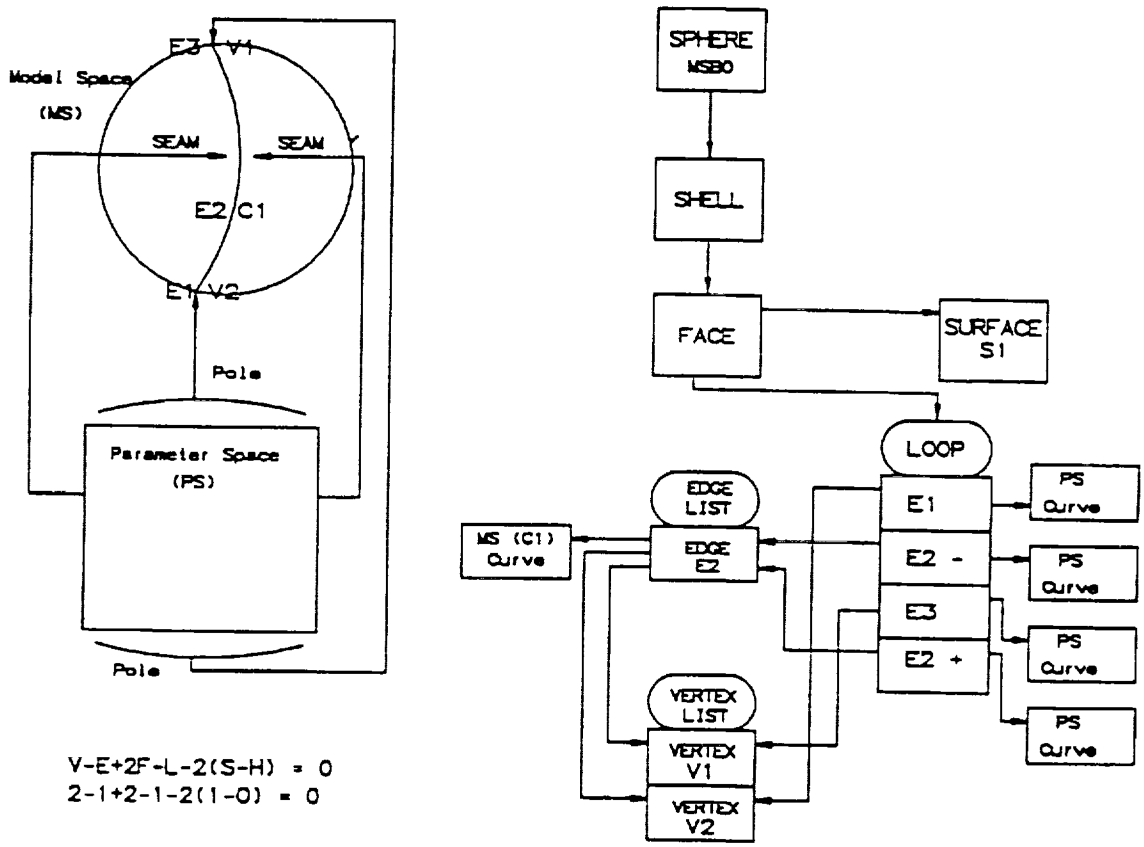

Figure 10: One of the possible MSBO representations and Euler formula of a sphere.

References

[1] Rockwell Automated Grid Generation System and Related Codes, NA-92-1482, North American Aircraft, Rockwell International Corporation, P.O. Box 92098, Los Angeles, CA 90009

[2] NASA Geometry Specification for Computational Fluid Dynamics.

[3] The Initial Graphics Exchange Specification (IGES), distributed by National Computer Graphics Association, Administrator, IGES/PDES Organization, 2722 Merrilee Drive, Suite 200, Fairfax, VA 22031.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1993

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
Superpatch

5. FUNDING NUMBERS
NAS2-12961

6. AUTHOR(S)
Jin J. Chou and Francis Enomoto

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Sciences Corporation
P.O. Box 390757
Mountain View, CA 94039

8. PERFORMING ORGANIZATION REPORT NUMBER
A-94010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-177627

11. SUPPLEMENTARY NOTES
Point of Contact: Matt Blake, Ames Research Center, MS T045-2, Moffett Field, CA 94035-1000
(415) 604-4978

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited
Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This document describes a design model geometry file format with topology information used for model geometry exchange in Computational Fluid Dynamics (CFD) grid generation. The first part of this document describes the reason why this file format is necessary and what we want to achieve through the file format. The second part provides the information stored in the file format. The purpose of the publication of this document is for broad dissemination of this file format and for the wide participation of testing.

14. SUBJECT TERMS
Boundary representation, Grid generation, Geometry

15. NUMBER OF PAGES
37

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		